Applying STLplus to GRACE Terrestrial Water Storage Data

Rizka Amelia Dwi Safira

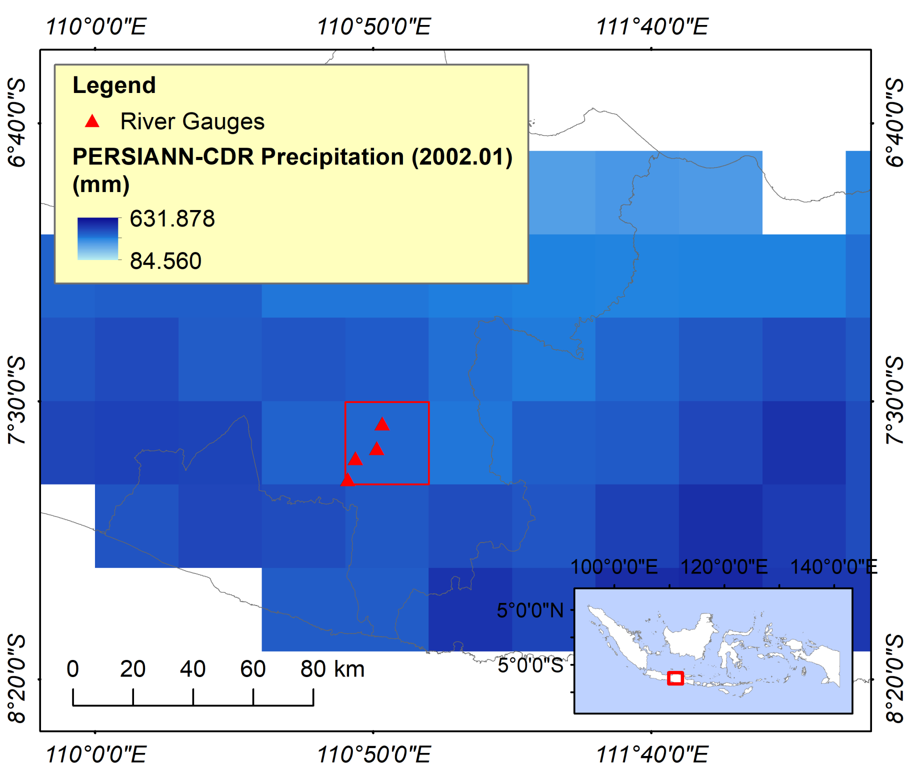
2025-02-19

This notebook aims to apply STLplus to GRACE/GRACE-FO Terrestrial Water Storage Anomaly (TWSA) data from CSR (Center for Space Research) with a 0.25-degree spatial resolution and a one-month temporal resolution. STLplus is an improved decomposition approach of seasonal and trend decomposition using Loess (STL) by Cleveland et al. (2019), developed by **Ryan Hafen**.

STLplus decomposes signals into trend, seasonal, and remainder constituents with a better treatment of missing values. Therefore, it can handle intra-gaps (within the GRACE mission from 2002.04 to 2017.05) and continuous inter-gaps (11 gap months between GRACE and GRACE-FO missions from 2017.06 to 2018.05) of GRACE TWSA data, with notable findings from previous studies (Arshad et al., 2024; Ali et al., 2024; Khorrami et al., 2023). We fill the missing TWSA values with the decomposed trend of TWSA and then add the average seasonal and residual components for the respective months.

Since we conduct a straightforward analysis, we will only select one grid data from GRACE CSR, with a latitude of -7.625 and a longitude of 110.875, located in Java Island of Indonesia. This selection is based on the availability of four river gauge observation data (the distribution is shown in the figure below) to validate our decomposition results.

Code reference: Harrington (2020)



```r
# Libraries
# install.packages("stlplus")
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(dplyr)
library(devtools)
```

```
## Loading required package: usethis
```

```r
library(stlplus)
library(ggfortify)
library(xts)
```

```
## Loading required package: zoo
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
##
##
## ######################### Warning from 'xts' package ##########################
## #                                                                             #
## # The dplyr lag() function breaks how base R's lag() function is supposed to  #
## # work, which breaks lag(my_xts). Calls to lag(my_xts) that you type or       #
## # source() into this session won't work correctly.                            #
## #                                                                             #
## # Use stats::lag() to make sure you're not using dplyr::lag(), or you can add #
## # conflictRules('dplyr', exclude = 'lag') to your .Rprofile to stop           #
## # dplyr from breaking base R's lag() function.                                #
## #                                                                             #
## # Code in packages is not affected. It's protected by R's namespace mechanism #
## # Set `options(xts.warn_dplyr_breaks_lag = FALSE)` to suppress this warning.  #
## #                                                                             #
## ###############################################################################
##
## Attaching package: 'xts'
##
## The following objects are masked from 'package:dplyr':
##
##     first, last
```

```r
library(lubridate)
```

Calling CSV Data

First, we call our GRACE TWSA (TWSA = LWE or liquid water equivalent in cm).

```
# Hydrology Data in CSV
df <- read.csv("D:\\01. RIZKA\\Repo__GitHub\\4_STLplus\\GRACE_CSR_2002.04_2024.09.csv",
               sep=",")
df$time <- as.Date(df$time)
head(df)
```

```
##         time     lon    lat     lwe_cm
## 1 2002-04-18 105.375 -8.625 -0.4274038
## 2 2002-04-18 111.625 -8.125 14.1238540
## 3 2002-04-18 111.625 -8.375  0.9586106
## 4 2002-04-18 111.625 -8.625  0.9586106
## 5 2002-04-18 111.375 -6.125  3.0323179
## 6 2002-04-18 111.375 -6.375  3.0323179
```

```
# Execute for one grid only (for simple practice)
df1 <- subset(df, lat == "-7.625" & lon == "110.875")
str(df1)
```

```
## 'data.frame':    237 obs. of  4 variables:
##  $ time  : Date, format: "2002-04-18" "2002-05-10" ...
##  $ lon   : num  111 111 111 111 111 ...
##  $ lat   : num  -7.62 -7.62 -7.62 -7.62 -7.62 ...
##  $ lwe_cm: num  16.12 12.1 -5.59 -7.91 -7.98 ...
```

```
# Check the start and end date of df1
head(df1)
```

```
##            time     lon    lat     lwe_cm
## 84   2002-04-18 110.875 -7.625  16.118933
## 491  2002-05-10 110.875 -7.625  12.098170
## 897  2002-08-16 110.875 -7.625  -5.594867
## 1304 2002-09-16 110.875 -7.625  -7.907865
## 1710 2002-10-16 110.875 -7.625  -7.984222
## 2118 2002-11-16 110.875 -7.625 -10.904664
```

```
tail(df1)
```

```
##             time     lon    lat     lwe_cm
## 94101 2024-04-16 110.875 -7.625 16.4697760
## 94508 2024-05-16 110.875 -7.625 10.0163690
## 94915 2024-06-16 110.875 -7.625  6.0334560
## 95322 2024-07-16 110.875 -7.625 -0.2065086
## 95729 2024-08-16 110.875 -7.625 -3.4315872
## 96134 2024-09-16 110.875 -7.625 -2.9167120
```

According to the information above, our TWSA data ranges from 2002.04 to 2024.09.

Now, we will check whether there are duplicate months. This check investigates whether there is more than one TWSA value in one month.

```
# Add date column with data format "Y-m-01" to check double values in a month
df1$date <- as.Date(format(df1$time, "%Y-%m-01"))
# Check duplicate
df1[duplicated(df1$date), ]
```

```
##             time     lon    lat     lwe_cm       date
```

```
## 44853 2011-10-31 110.875 -7.625 -5.759689 2011-10-01
## 58690 2015-04-27 110.875 -7.625  8.062920 2015-04-01
```

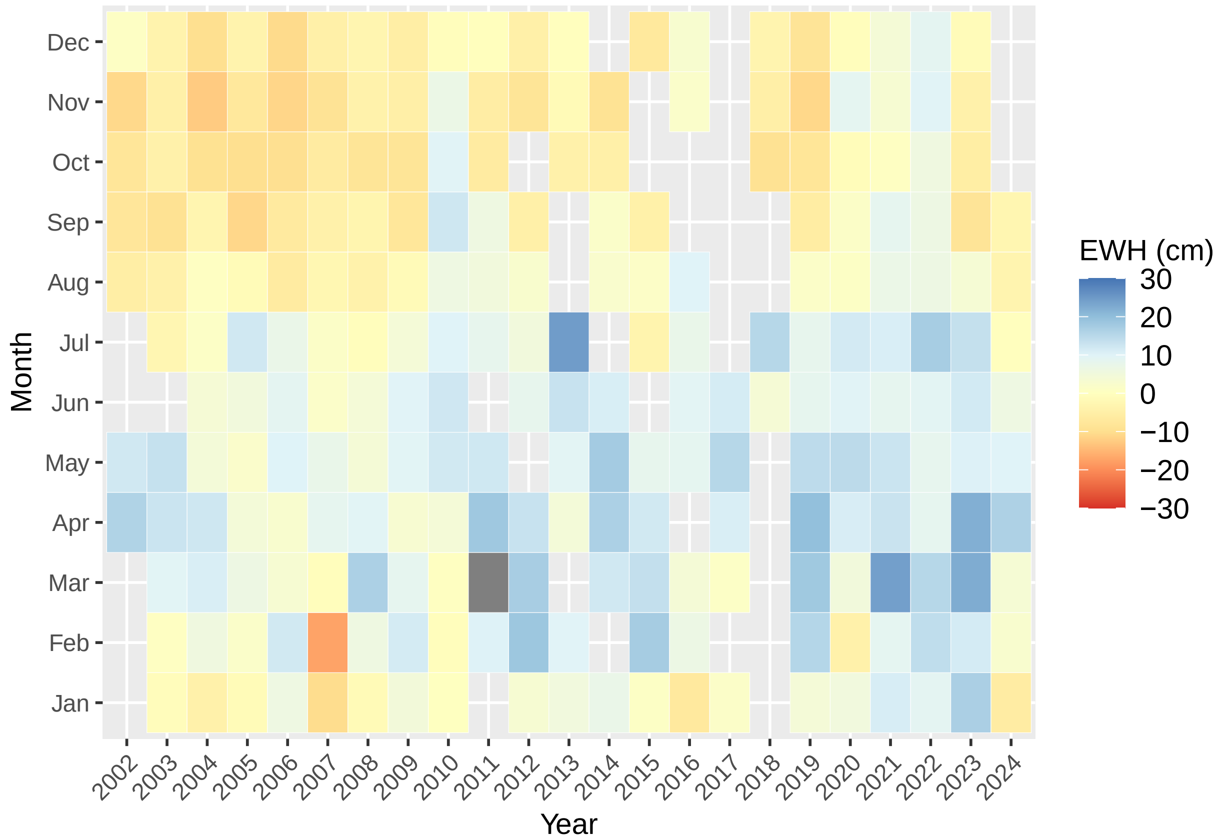
We can see that there are two values in 2011.10 and 2015.04. Referring to the GRACE & GRACE-FO Data Months/Days Table, there were no missing values in 2011.11 and 2015.05 or one month after 2011.10 and 2015.04. Therefore, we will assign the "date" column in rows with 44853 and 58690 indexes as 2011.11.01 and 2015.05.01.

```
df1$date[duplicated(df1$date)] <- df1$date[duplicated(df1$date)] %m+% months(1)
```

Visualizing Missing Data

Next, we will visualize the gaps. Firstly, we extract the month and year in separate columns.

```
# Visualize missing data
df1$month <- as.character(format(df1$date, "%m"))
df1$month <- factor(
  month.abb[as.numeric(df1$month)],
  levels = month.abb)
df1$year <- as.character(format(df1$time, "%Y"))
# Tile
ggplot(df1, aes(year, month))+
  geom_tile(aes(fill = lwe_cm), color = "white") +
  scale_fill_distiller(palette = "RdYlBu", direction = 1, limits = c(-30,30),
                       name = "EWH (cm)") +
  labs(x = "Year", y = "Month") +
  theme(
    #panel.grid = element_blank(),
    legend.text = element_text(size = 11),
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.position = "right"
  )
```



Subsequently, we will also visualize the time series graph and look at the missing data. Firstly, we create a new data frame with a sequential date from 2002.04 to 2024.09. Secondly, we merge the new data frame with our TWSA data. This process aims to make the missing values to be visible.

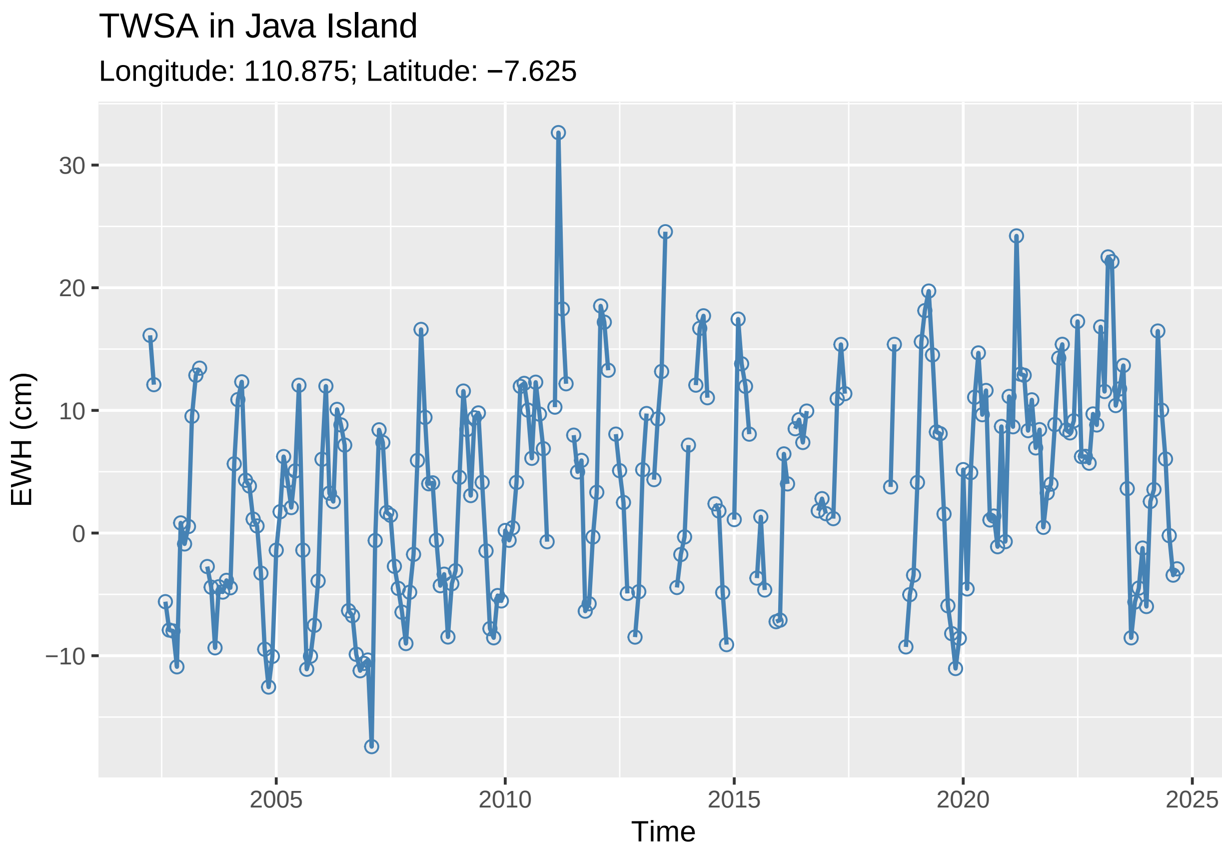
```
# Make dummy DataFrame
start_date <- as.Date("2002-04-01")
end_date <- as.Date("2024-09-30")
date_seq <- seq.Date(from = start_date, to = end_date, by = "month")
date_seq <- tibble(date = date_seq)

# Merge
df2 <- left_join(x = date_seq, y = df1, by = "date" )
str(df2)
```

```
## tibble [270 x 7] (S3: tbl_df/tbl/data.frame)
##  $ date  : Date[1:270], format: "2002-04-01" "2002-05-01" ...
##  $ time  : Date[1:270], format: "2002-04-18" "2002-05-10" ...
##  $ lon   : num [1:270] 111 111 NA NA 111 ...
##  $ lat   : num [1:270] -7.62 -7.62 NA NA -7.62 ...
##  $ lwe_cm: num [1:270] 16.12 12.1 NA NA -5.59 ...
##  $ month : Factor w/ 12 levels "Jan","Feb","Mar",..: 4 5 NA NA 8 9 10 11 12 1 ...
##  $ year  : chr [1:270] "2002" "2002" NA NA ...
```

```
# Time-series
df2 %>% ggplot(aes(date, lwe_cm))+
  geom_line(linewidth=.75,color ="steelblue")+
  geom_point(size=2,shape=1,color="steelblue")+
  labs(title="TWSA in Java Island",
```

```
       subtitle = "Longitude: 110.875; Latitude: -7.625",
       x= "Time", y="EWH (cm)")
```

```
## Warning: Removed 33 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



Performing STLplus

Before we perform the STLplus, first, we create the time series object. We use a frequency of 12 since our data is monthly.

```
df2_ts <- ts(df2$lwe_cm,
             start = c(2002, 4),
             frequency = 12)
df2_ts
```

```
##              Jan         Feb         Mar         Apr         May         Jun
## 2002                                      16.1189330  12.0981700          NA
## 2003  -0.8876171   0.5331813   9.5234995  12.8622360  13.4414390          NA
## 2004  -4.4625880   5.6406946  10.8759400  12.3303220   4.3048205   3.8356774
## 2005  -1.3965293   1.7385664   6.2345366   4.2758436   2.0772336   5.0651064
## 2006   6.0152264  11.9791240   3.2499110   2.5685003  10.0817150   8.8176380
## 2007 -10.3530380 -17.4117970  -0.6118972   8.4167280   7.3825570   1.6787654
## 2008  -1.7371503   5.9180403  16.6004700   9.4270820   4.0179950   4.0918820
## 2009   4.5468445  11.5751220   8.4334060   3.0549395   9.3966390   9.7934220
## 2010   0.2081365  -0.5919407   0.4267310   4.1294300  11.9578070  12.1993885
## 2011          NA  10.2649300  32.6476550  18.2715260  12.1712640          NA
```

```
## 2012   3.3243077  18.5173630  17.1922100  13.2719170          NA   8.0645360
## 2013   5.1628020   9.7432390          NA   4.3550880   9.3057140  13.1715600
## 2014   7.1654468          NA  12.0605380  16.6881260  17.7105450  11.0332710
## 2015   1.0961028  17.4413280  13.7991820  11.9598130   8.0629200          NA
## 2016  -7.0959096   6.4513516   4.0084534          NA   8.5080100   9.2415180
## 2017   1.5812551          NA   1.1773814  10.9504220  15.3759760  11.3641050
## 2018          NA          NA          NA          NA          NA   3.7602340
## 2019   4.1258040  15.5991400  18.1304970  19.7273040  14.5234100   8.2453330
## 2020   5.1719150  -4.5548390   4.9236894  11.0716710  14.6850150   9.6485990
## 2021  11.1355690   8.6664780  24.2233900  12.9559240  12.8658090   8.3499900
## 2022   8.8469070  14.2659235  15.3851880   8.4045530   8.1669035   9.1505200
## 2023  16.8113120  11.5398340  22.5131910  22.1252690  10.3941510  11.7525215
## 2024  -5.9890020   2.5720768   3.5432482  16.4697760  10.0163690   6.0334560
##              Jul         Aug         Sep         Oct         Nov         Dec
## 2002          NA  -5.5948668  -7.9078655  -7.9842215 -10.9046640   0.8269757
## 2003  -2.7306347  -4.4137664  -9.3591820  -4.3797016  -4.8119650  -3.8593836
## 2004   1.1334212   0.5765046  -3.2716177  -9.4725600 -12.5565540 -10.0543940
## 2005  12.0501430  -1.3945452 -11.1023390 -10.0389670  -7.5224023  -3.9072304
## 2006   7.1725360  -6.3284183  -6.7384477  -9.8853780 -11.2242155 -10.6273140
## 2007   1.4452041  -2.7083833  -4.5148926  -6.4516940  -9.0079280  -4.8265760
## 2008  -0.5935978  -4.2902300  -3.3544762  -8.4724740  -4.1311820  -3.0731820
## 2009   4.1335454  -1.4577413  -7.7893643  -8.5393460  -5.0963254  -5.5313296
## 2010  10.0235140   6.0924582  12.2967400   9.6955270   6.8763995  -0.7023067
## 2011   7.9810796   4.9870296   5.9211698  -6.3675847  -5.7596890  -0.3206960
## 2012   5.0854726   2.4933019  -4.9200134          NA  -8.4816060  -4.7842784
## 2013  24.5650400          NA          NA  -4.4351810  -1.7511829  -0.3141029
## 2014          NA   2.3859656   1.7986898  -4.8486840  -9.0956640          NA
## 2015  -3.6724334   1.3231641  -4.6426750          NA          NA  -7.2238240
## 2016   7.3843390   9.9446530          NA          NA   1.8113686   2.8052857
## 2017          NA          NA          NA          NA          NA          NA
## 2018  15.3834250          NA          NA  -9.2833460  -5.0221744  -3.4309442
## 2019   8.1010760   1.5557609  -5.9271550  -8.1975470 -11.0433650  -8.5838810
## 2020  11.6225195   1.0666606   1.3861967  -1.1136476   8.6886850  -0.6978874
## 2021  10.8591760   6.9412136   8.4307260   0.4650341   3.2730873   3.9915636
## 2022  17.2544200   6.2311640   6.2514467   5.6953983   9.7086990   8.8149560
## 2023  13.6608700   3.6195254  -8.5422140  -5.6219540  -4.4889870  -1.2251600
## 2024  -0.2065086  -3.4315872  -2.9167120
```

Next, we perform the STLplus and extract the decomposed signals into a new table.

```
# Perform STL
# stlobj <- stlplus(df2_ts, s.window=12, s.degree=2,
#                    t.window=12, t.degree=2, fc.window=12)
stlobj <- stlplus(df2_ts, s.window="periodic")

# Extract decomposed signals
trend <- trend(stlobj)
seasonal <- seasonal(stlobj)
remainder <- remainder(stlobj)

# Mutate in df
df3 <- df2 %>%
  mutate(trend = trend,
         seasonal = seasonal,
         remainder = remainder)
```

```
head(df3)
```

```
## # A tibble: 6 x 10
##   date       time         lon   lat lwe_cm month year  trend seasonal remainder
##   <date>     <date>     <dbl> <dbl>  <dbl> <fct> <chr> <dbl>    <dbl>     <dbl>
## 1 2002-04-01 2002-04-18  111. -7.62  16.1  Apr   2002   3.83     7.56      4.73
## 2 2002-05-01 2002-05-10  111. -7.62  12.1  May   2002   3.44     6.61      2.04
## 3 2002-06-01 NA           NA  NA     NA    <NA>  <NA>   3.07     4.39     NA
## 4 2002-07-01 NA           NA  NA     NA    <NA>  <NA>   2.74     3.87     NA
## 5 2002-08-01 2002-08-16  111. -7.62  -5.59 Aug   2002   2.42    -2.70     -5.32
## 6 2002-09-01 2002-09-16  111. -7.62  -7.91 Sep   2002   2.06    -5.86     -4.11
```

Further, we calculate the average monthly values of seasonal and remainder components. Firstly, we need to refill the month column including rows with NAs.

```
# Calculate mean for seasonal and remainder signals
df3$month <- as.character(format(df3$date, "%m"))
df3$month <- factor(
  month.abb[as.numeric(df3$month)],
  levels = month.abb)
df3 <- df3 %>%
  group_by(month) %>%
  mutate(meanSeas = mean(seasonal, na.rm = TRUE),
          meanRema = mean(remainder, na.rm = TRUE))
head(df3)
```

```
## # A tibble: 6 x 12
## # Groups:   month [6]
##   date       time         lon   lat lwe_cm month year  trend seasonal remainder
##   <date>     <date>     <dbl> <dbl>  <dbl> <fct> <chr> <dbl>    <dbl>     <dbl>
## 1 2002-04-01 2002-04-18  111. -7.62  16.1  Apr   2002   3.83     7.56      4.73
## 2 2002-05-01 2002-05-10  111. -7.62  12.1  May   2002   3.44     6.61      2.04
## 3 2002-06-01 NA           NA  NA     NA    Jun   <NA>   3.07     4.39     NA
## 4 2002-07-01 NA           NA  NA     NA    Jul   <NA>   2.74     3.87     NA
## 5 2002-08-01 2002-08-16  111. -7.62  -5.59 Aug   2002   2.42    -2.70     -5.32
## 6 2002-09-01 2002-09-16  111. -7.62  -7.91 Sep   2002   2.06    -5.86     -4.11
## # i 2 more variables: meanSeas <dbl>, meanRema <dbl>
```

Filling the Missing Month Values

We fill in the missing month values using the trend, mean seasonal, and mean remainder values.

```
# Which na?
missing_indices <- which(is.na(df3$lwe_cm))
missing_indices
```

```
##  [1]   3   4  15 106 111 122 127 132 137 138 143 148 153 159 163 164 169 174 175
## [20] 179 184 185 186 187 188 189 190 191 192 193 194 197 198
```

```
length(missing_indices)
```

```
## [1] 33
```

We know that there are 33 missing values from 2002.04 to 2024.09.

```
# Fill in
reconstruct_stl <- df3$trend + df3$meanSeas + df3$meanRema
df3_recon <- df3 %>%
```

```
  ungroup() %>%  # Remove grouping to ensure global replacement
  mutate(lwe_cm = replace(lwe_cm, is.na(lwe_cm), reconstruct_stl[missing_indices]))
head(df3_recon)
```

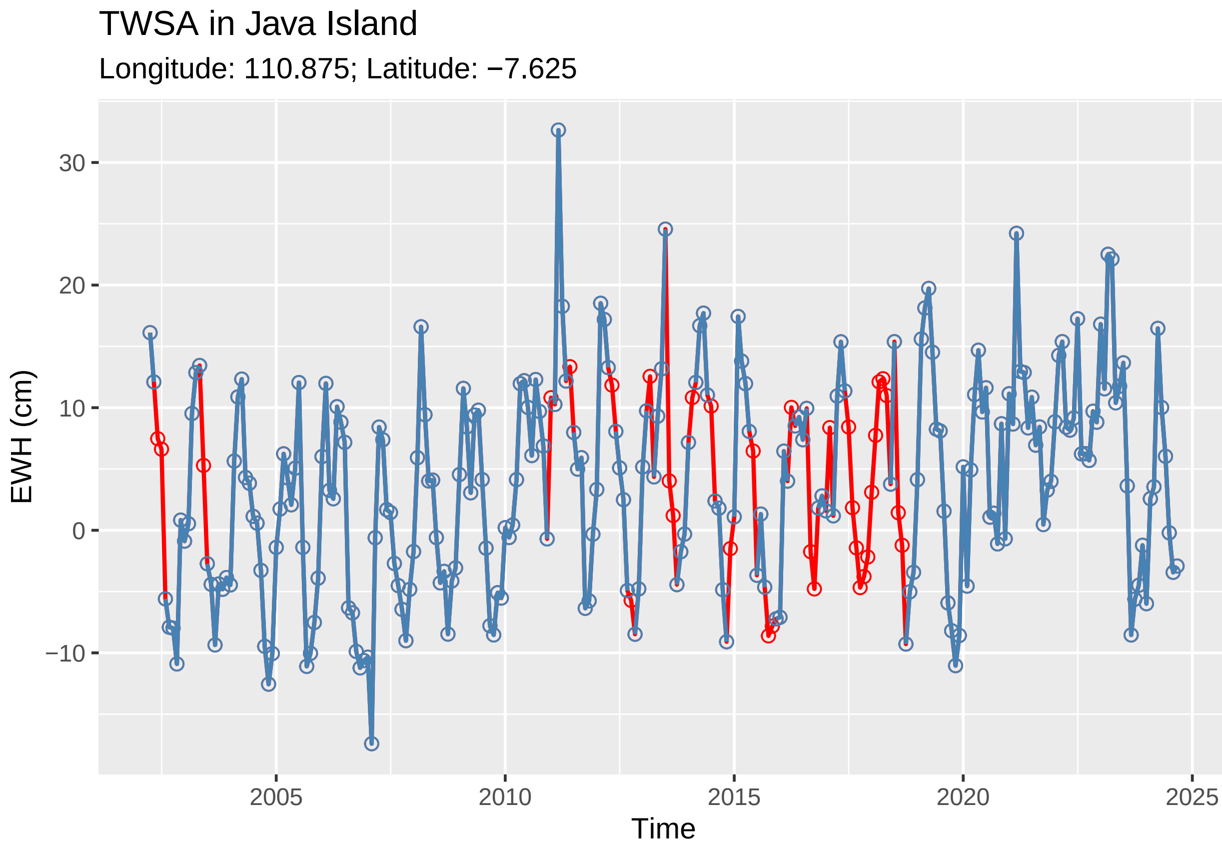
```
## # A tibble: 6 x 12
##   date       time         lon   lat lwe_cm month year  trend seasonal remainder
##   <date>     <date>     <dbl> <dbl>  <dbl> <fct> <chr> <dbl>    <dbl>     <dbl>
## 1 2002-04-01 2002-04-18  111. -7.62  16.1  Apr   2002   3.83     7.56      4.73
## 2 2002-05-01 2002-05-10  111. -7.62  12.1  May   2002   3.44     6.61      2.04
## 3 2002-06-01 NA            NA  NA     7.47 Jun   <NA>   3.07     4.39     NA
## 4 2002-07-01 NA            NA  NA     6.61 Jul   <NA>   2.74     3.87     NA
## 5 2002-08-01 2002-08-16  111. -7.62  -5.59 Aug   2002   2.42    -2.70     -5.32
## 6 2002-09-01 2002-09-16  111. -7.62  -7.91 Sep   2002   2.06    -5.86     -4.11
## # i 2 more variables: meanSeas <dbl>, meanRema <dbl>
```

The missing LWEs have been successfully filled in. Next, we will visualize the continuous time series.

```
# Visualize
ggplot()+
  geom_line(data = df3_recon, aes(x=date, y=lwe_cm), size=.75,color ="red")+
  geom_point(data = df3_recon, aes(x=date, y=lwe_cm), size=2,shape=1,color="red")+
  geom_line(data = df3, aes(x=date,y=lwe_cm),size=.75,color="steelblue")+
  geom_point(data = df3, aes(x=date, y=lwe_cm), size=2,shape=1,color="steelblue")+
  labs(title="TWSA in Java Island",
       subtitle = "Longitude: 110.875; Latitude: -7.625", x= "Time", y="EWH (cm)")
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning: Removed 33 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



Validation

Further, we validate the original and filled LWE values using the monthly river gauge data. The data recorded in each gauge represents the water level data (in meters). Therefore, we assume water level data as the surface water component in terrestrial water storage. We use the average water level from four gauges to generate the final value for each month. Finally, we subtract the monthly water level value by the 2004.01 to 2009.12 average water level value to compute its anomaly, making it comparable with GRACE TWSA data.

```r
# validate stl with TMAA
TMAA <- read.csv("D:\\01. RIZKA\\Repo__GitHub\\4_STLplus\\WaterLevelData.csv",
                 sep=";")
TMAA$date <- as.Date(TMAA$date)
head(TMAA)
```

```
##         date        WLA
## 1 2003-01-01 0.66977623
## 2 2003-02-01 1.49644290
## 3 2003-03-01 1.41310957
## 4 2003-04-01 0.31310957
## 5 2003-05-01 0.24144290
## 6 2003-06-01 0.08810957
```

```r
# Merge with df3_recon
df3_validate <- left_join(df3_recon[,c("date","lwe_cm")], TMAA, by = "date")
df3_validate
```

```
## # A tibble: 270 x 3
##    date       lwe_cm    WLA
```

```
##    <date>       <dbl>  <dbl>
##  1 2002-04-01  16.1   NA
##  2 2002-05-01  12.1   NA
##  3 2002-06-01   7.47  NA
##  4 2002-07-01   6.61  NA
##  5 2002-08-01  -5.59  NA
##  6 2002-09-01  -7.91  NA
##  7 2002-10-01  -7.98  NA
##  8 2002-11-01 -10.9   NA
##  9 2002-12-01   0.827 NA
## 10 2003-01-01  -0.888  0.670
## # i 260 more rows
```

We will shift the WLA to several months early and check the highest coefficient values.

```
# Shift WLA
df3_validate <- df3_validate %>%
  mutate(WLA_1 = lag(WLA, n=1),
         WLA_2 = lag(WLA, n=2),
         WLA_3 = lag(WLA, n=3),
         WLA_4 = lag(WLA, n=4),
         WLA_5 = lag(WLA, n=5))
df3_validate
```

```
## # A tibble: 270 x 8
##    date        lwe_cm    WLA WLA_1 WLA_2 WLA_3 WLA_4 WLA_5
##    <date>       <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1 2002-04-01  16.1   NA        NA    NA    NA    NA    NA
##  2 2002-05-01  12.1   NA        NA    NA    NA    NA    NA
##  3 2002-06-01   7.47  NA        NA    NA    NA    NA    NA
##  4 2002-07-01   6.61  NA        NA    NA    NA    NA    NA
##  5 2002-08-01  -5.59  NA        NA    NA    NA    NA    NA
##  6 2002-09-01  -7.91  NA        NA    NA    NA    NA    NA
##  7 2002-10-01  -7.98  NA        NA    NA    NA    NA    NA
##  8 2002-11-01 -10.9   NA        NA    NA    NA    NA    NA
##  9 2002-12-01   0.827 NA        NA    NA    NA    NA    NA
## 10 2003-01-01  -0.888  0.670    NA    NA    NA    NA    NA
## # i 260 more rows
```

```
# Calculate correlation
corr <- cor(df3_validate[,c(2:8)], use="pairwise.complete.obs")
corr
```

```
##           lwe_cm         WLA       WLA_1     WLA_2     WLA_3       WLA_4
## lwe_cm 1.0000000  0.42190313  0.63562574 0.6667726 0.6035481  0.42382858
## WLA    0.4219031  1.00000000  0.74600898 0.4899319 0.1645846 -0.09109368
## WLA_1  0.6356257  0.74600898  1.00000000 0.7460090 0.4899319  0.16458458
## WLA_2  0.6667726  0.48993188  0.74600898 1.0000000 0.7460090  0.48993188
## WLA_3  0.6035481  0.16458458  0.48993188 0.7460090 1.0000000  0.74600898
## WLA_4  0.4238286 -0.09109368  0.16458458 0.4899319 0.7460090  1.00000000
## WLA_5  0.1660119 -0.29112758 -0.09109368 0.1645846 0.4899319  0.74600898
##              WLA_5
## lwe_cm  0.16601194
## WLA    -0.29112758
## WLA_1  -0.09109368
## WLA_2   0.16458458
```

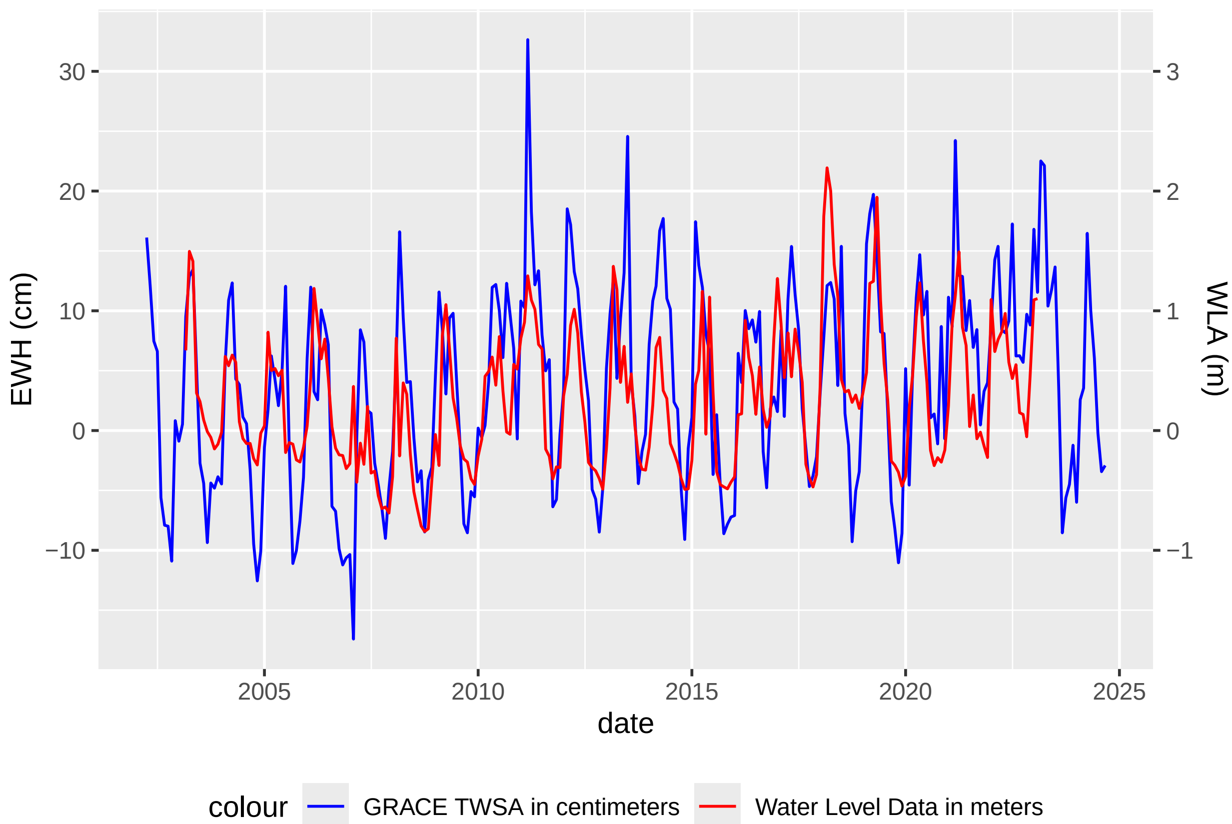
```
## WLA_3   0.48993188
## WLA_4   0.74600898
## WLA_5   1.00000000
```

The results above imply that the highest correlation between WLA and GRACE TWSA happens at the 2-month lag of WLA with a correlation of **0.667.**

Next, we visualize it.

```
# Visualize
ggplot(df3_validate, aes(x = date)) +
  geom_line(aes(y = lwe_cm, color = "GRACE TWSA in centimeters")) +
  geom_line(aes(y = WLA_2 * 10, color = "Water Level Data in meters")) +
  scale_y_continuous(
    name = "EWH (cm)",
    sec.axis = sec_axis(~ . / 10, name = "WLA (m)")
  ) +
  scale_color_manual(values = c("GRACE TWSA in centimeters" = "blue",
                                "Water Level Data in meters" = "red"))+
  theme(legend.position = "bottom")
```

```
## Warning: Removed 30 rows containing missing values or values outside the scale range
## (`geom_line()`).
```



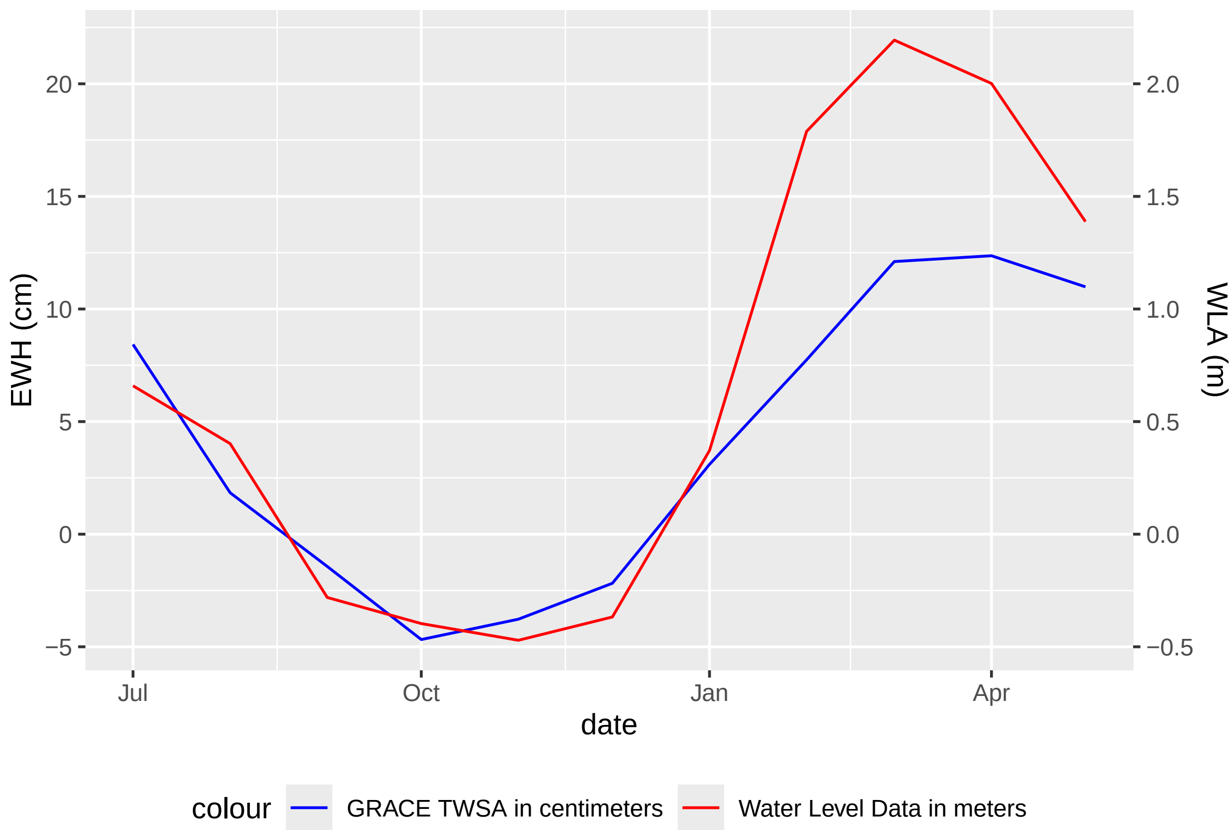
The difference in amplitudes between both datasets is possible since WLA only accounts for one of TWSA components (it does not cover soil moisture storage, plant canopy storage, snow water equivalent, and groundwater storage components in TWSA).

Next, we will look at closer at the longest gap months between GRACE and GRACE-FO missions and how the STLplus approach performs.

```
# Subset only at 11 month gaps
longest_gaps <- subset(df3_validate,
                       date >= "2017-07-01" & date <= "2018-05-01")
longest_gaps
```

```
## # A tibble: 11 x 8
##    date       lwe_cm    WLA  WLA_1  WLA_2  WLA_3  WLA_4  WLA_5
##    <date>      <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
##  1 2017-07-01   8.43 -0.281  0.402  0.659  0.847  0.449  0.816
##  2 2017-08-01   1.84 -0.397 -0.281  0.402  0.659  0.847  0.449
##  3 2017-09-01  -1.43 -0.471 -0.397 -0.281  0.402  0.659  0.847
##  4 2017-10-01  -4.68 -0.368 -0.471 -0.397 -0.281  0.402  0.659
##  5 2017-11-01  -3.78  0.371 -0.368 -0.471 -0.397 -0.281  0.402
##  6 2017-12-01  -2.18  1.79   0.371 -0.368 -0.471 -0.397 -0.281
##  7 2018-01-01   3.10  2.19   1.79   0.371 -0.368 -0.471 -0.397
##  8 2018-02-01   7.74  2.00   2.19   1.79   0.371 -0.368 -0.471
##  9 2018-03-01  12.1   1.39   2.00   2.19   1.79   0.371 -0.368
## 10 2018-04-01  12.4   1.12   1.39   2.00   2.19   1.79   0.371
## 11 2018-05-01  11.0   0.427  1.12   1.39   2.00   2.19   1.79
```

```
# Visualize
ggplot(longest_gaps, aes(x = date)) +
  geom_line(aes(y = lwe_cm, color = "GRACE TWSA in centimeters")) +
  geom_line(aes(y = WLA_2 * 10, color = "Water Level Data in meters")) +
  scale_y_continuous(
    name = "EWH (cm)",
    sec.axis = sec_axis(~ . / 10, name = "WLA (m)")
  ) +
  scale_color_manual(values = c("GRACE TWSA in centimeters" = "blue",
                               "Water Level Data in meters" = "red"))+
  theme(legend.position = "bottom")
```




```r
# Calculate correlation
corr_longest_gaps <- cor(longest_gaps[,c("lwe_cm","WLA_2")])
corr_longest_gaps
```

```
##           lwe_cm     WLA_2
## lwe_cm 1.0000000 0.9464656
## WLA_2  0.9464656 1.0000000
```

The correlation analysis shows that by adding the mean seasonal and remainder components to the trend signals from the STLplus method, we can successfully overcome the missing gaps within the GRACE/GRACE-FO mission, resulting in a continuous time series.

Reference

- Ali, S., Ran, J., Luan, Y., Khorrami, B., Xiao, Y., & Tangdamrongsub, N. (2024). The GWR model-based regional downscaling of GRACE/GRACE-FO derived groundwater storage to investigate local-scale variations in the North China Plain. Science of The Total Environment, 908, 168239. https://doi.org/10.1016/j.scitotenv.2023.168239.
- Arshad, A., Mirchi, A., Taghvaeian, S., & AghaKouchak, A. (2024). Downscaled-GRACE data reveal anthropogenic and climate-induced water storage decline across the Indus Basin. Water Resources Research, 60(7), e2023WR035882. https://doi.org/10.1029/2023WR035882.
- Cleveland, R. B., Cleveland, W. S., McRae, J. E., & Terpenning, I. (1990). STL: A seasonal-trend decomposition. J. off. Stat, 6(1), 3-73.
- Hafen, R. stlplus. GitHub. Available at: https://github.com/hafen/stlplus.
- Harrington, M. (2020). Fill in Missing Cyclical Data using STL and Cross Validation. Available at:

https://www.mattrharrington.com/post/fill-in-missing-cyclical-data-using-seasonal-trend-loess-and-cross-validation.

- Khorrami, B., Ali, S., Sahin, O. G., & Gunduz, O. (2023). Model-coupled GRACE-based analysis of hydrological dynamics of drying Lake Urmia and its basin. Hydrological Processes, 37(5), e14893. https://doi.org/10.1002/hyp.14893.